Oracle SOA Suite 11g
Oracle SOA Suite 11g HL7 Inbound Customized HL7 Message Structure and Data Validation

michael@czapski.id.au
June 2010

Table of Contents

Introduction

Messages we used in previous articles dealing with HL7 Inbound were not strictly speaking valid, according to the default HL7 V2 ADT A01 message specification produced by the Oracle B2b Document Editor. Both the message structure was not quite right and the data was not quite right. To allow such messages in, we disabled Validation property in the B2B Trading Partnership Agreement.

In this article we will create a customized HL7 v2 ADT A01 structure which will allow us to successfully validate incoming messages. We will then modify the document definition and Partnership Agreements to use this custom structure and validate messages as they come in.

The customization we will discuss here only scratches the surface of what is possible with the Oracle B2B Document Editor.

Pre-requisites

It is assumed that a Windows XP SP3 platform with the Oracle SOA Suite 11g, installed and configured as discussed in "Installing Oracle SOA Suite for HL7 Exploration", published at http://blogs.czapski.id.au/wp-content/uploads/2010/06/01_Installing_Oracle_SOA_Suite_for_HL7_exploration_v1.1.pdf, is available and will be used for the work discussed in this article.

It is assumed that the HL7 solutions, discussed in blog articles Oracle SOA Suite 11g HL7 Inbound Example and Oracle SOA Suite 11g HL7 Inbound Example – Functional ACK Addendum are built and deployed.

It is assumed that data in the archive, HL7_messages_sources.zip, has been unpacked to C:\hl7\adt\data\. This archive is available from http://blogs.czapski.id.au/wp-content/uploads/2010/06/HL7_messages_sources.zip.

It is assumed that the free HL7 Browser tool, "HL7 Browser 1.0", available from the its author's page at http://mac.softpedia.com/developer/Michael-Litherland-5914.html, is available.

HL7 v2 Receiver Solution

The solution we will be modifying is a HL7 Receiver, which will receive v2 delimited ADT A01 messages and will write them to files in the file system. This solution was discussed and built in the blog article Oracle SOA Suite 11g HL7 Inbound Example – Functional ACK Addendum.

The solution consists of a B2B Listener Channel, to which HL7 v2 ADT A01 messages will be sent (B2b Layer) , and a SOA Composite which will receive these messages and will write them to a file in the file system (SOA Layer).
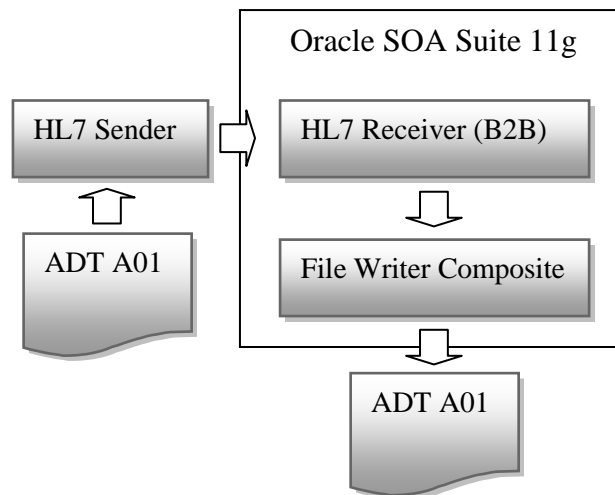
Messages in the sample message set use the following identifiers:

|	
Sending Application	SystemA
Sending Facility	HosA
Receiving Application	PI
Receiving Facility	MDM

Data Analysis and Guideline Customization

The Oracle B2B Document Editor includes Data Analysis instrumentation which allows analysis and validation of data using the Guideline (Message Structure). We will use this functionality to analyze the sample message and to customize the message structure so that it succeeds in validating our data.
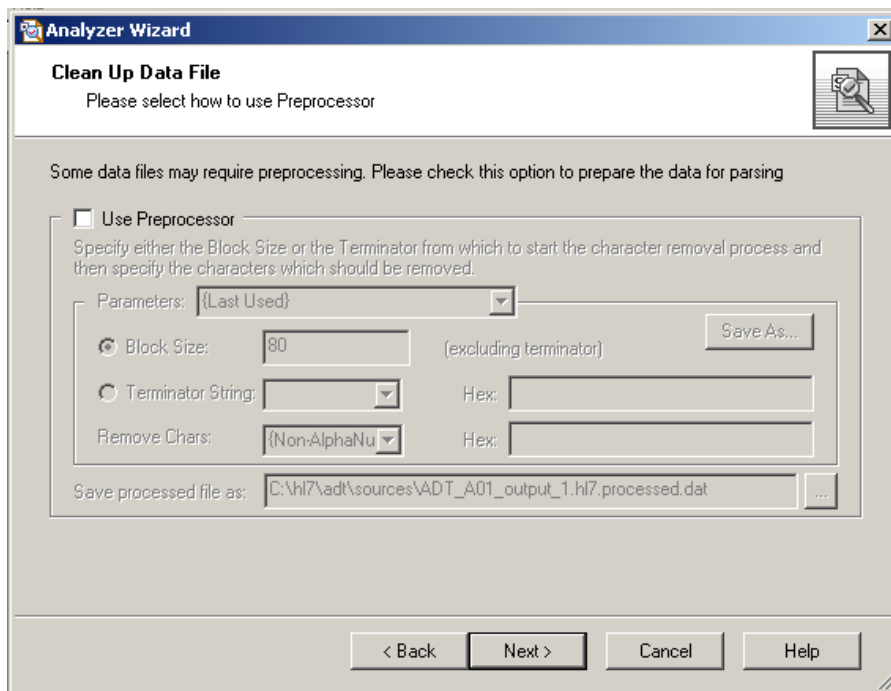
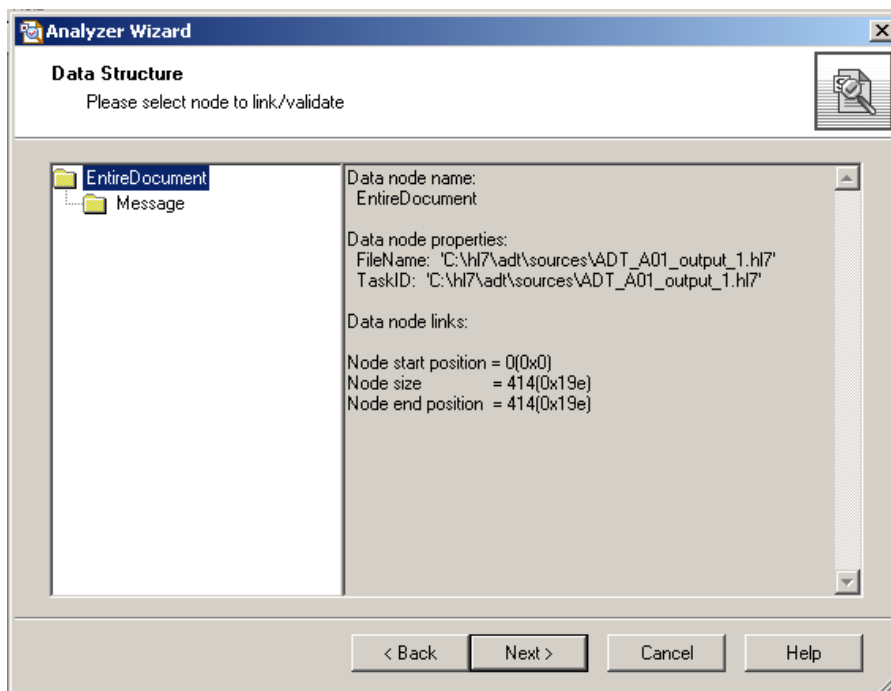Start the B2B Document Editor, which should be accessible through Start menu → Programs → Oracle as Oracle B2B.
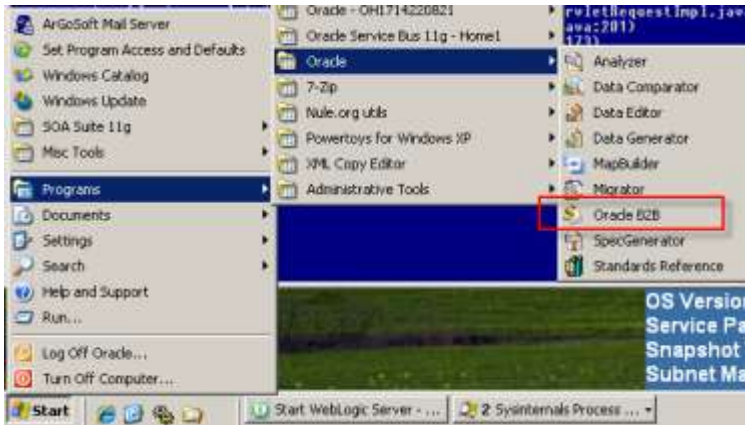
Click Analyzer and locate the data file C:\hl7\adt\sources\ADT_A01_output_1.hl7. Specify HL7 as data file type, then click Next.
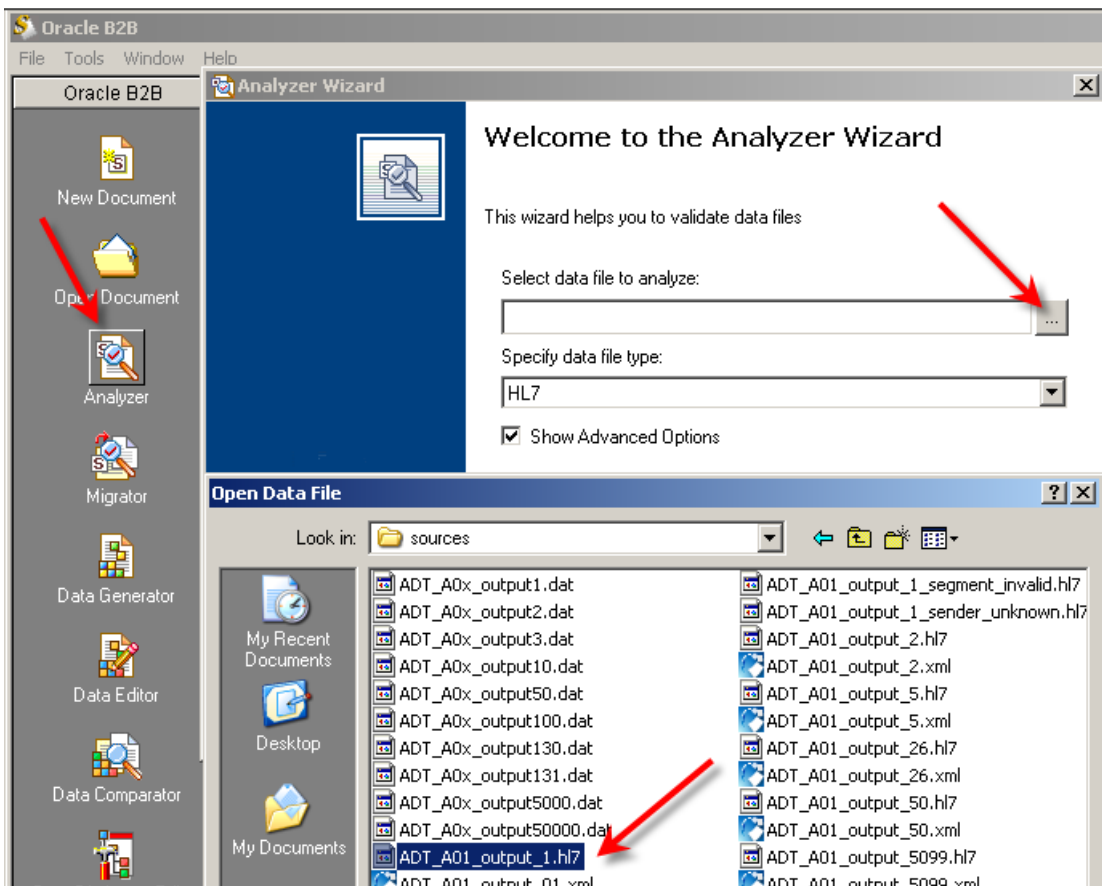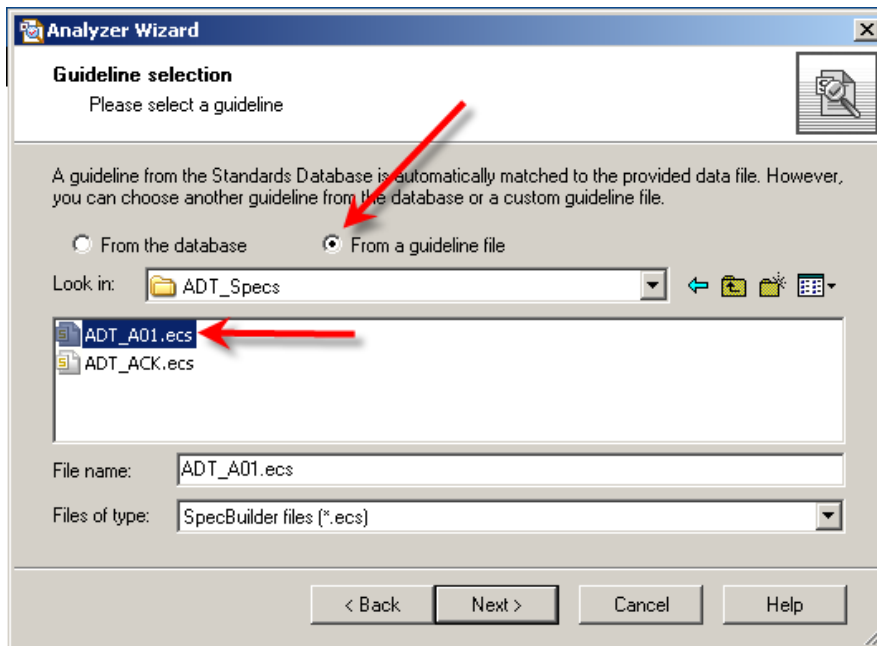
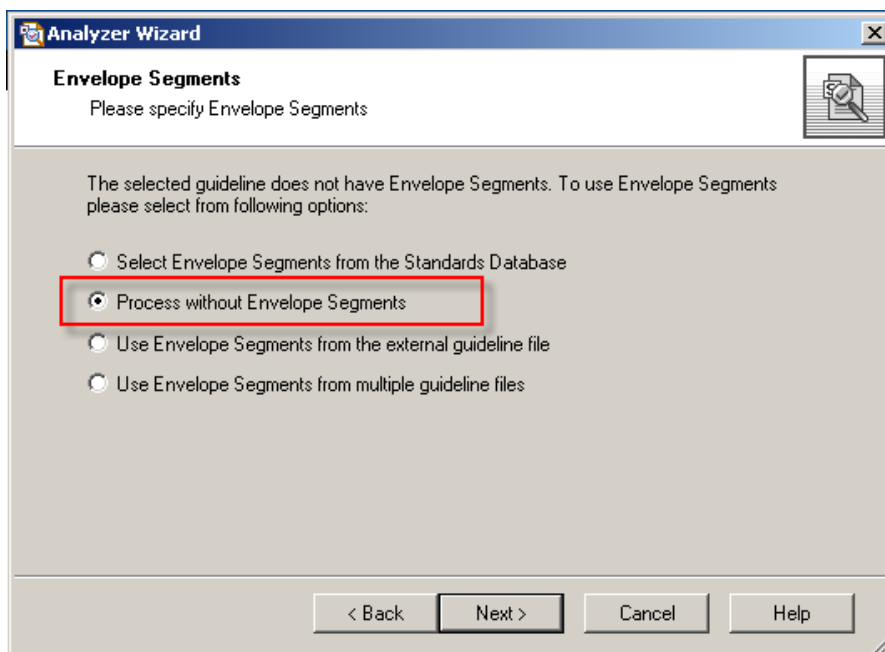Click Next to "Clean Up Data File".
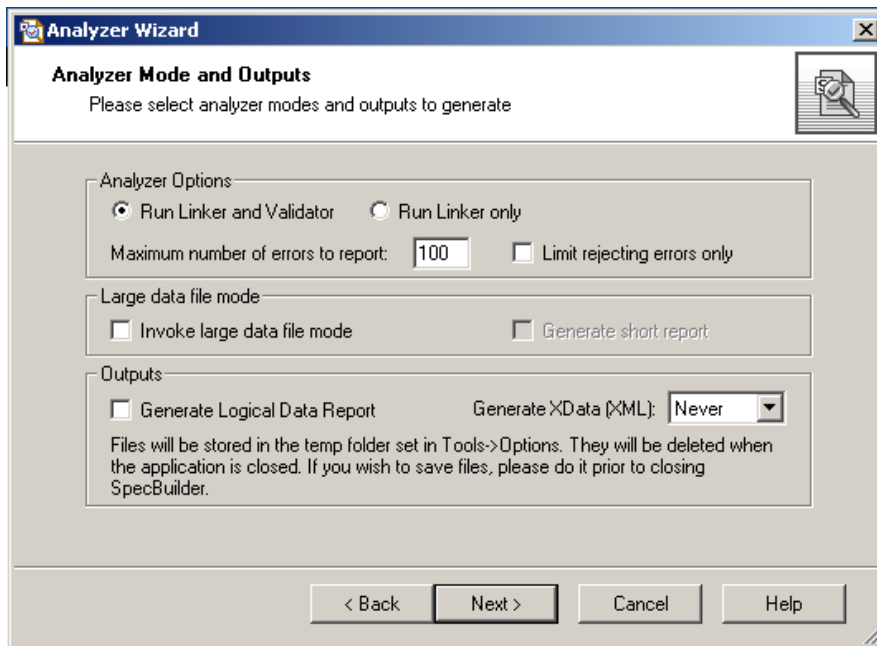

Click Next to "Data Structure".

Make sure to select the "From Guideline File" radio button and locate the ADT_A01.ecs guideline file, c:\hl7\adt\adt_specs\adt_a01.ecs, then click Next.
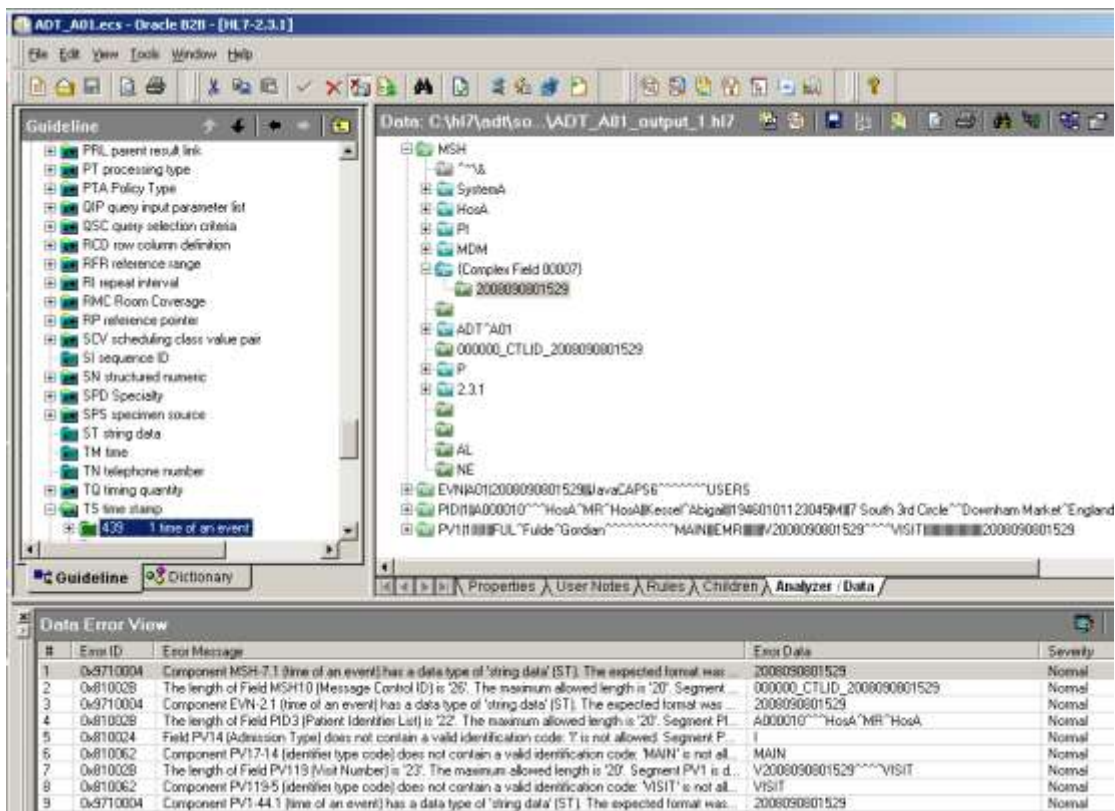
Accept default choice for "Envelope Segments" and click Next.


Accept default for "Analyzer Mode and Outputs" and click Next.

The data file will be analyzed and issues will be reported. The guideline structure appears in the top left hand side pane, the data file in the top right hand pane and the issues, if any, in the bottom pane.

Before going any further let's pull down the file menu and choose Save As to save a copy of the guideline. Let's name it ADT_A01_custom.ecs.
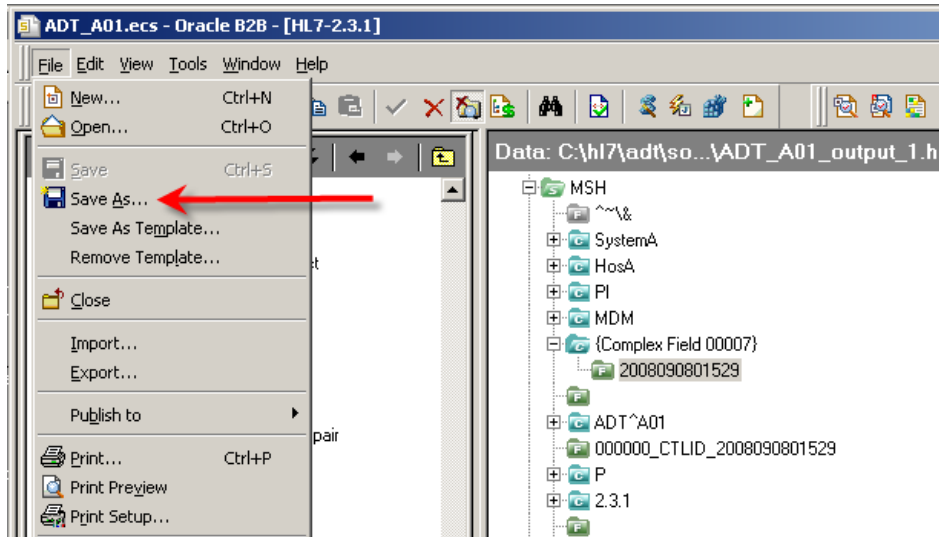
Let's double click on the first line in the Data Error View pane. The data type applicable to the error is selected in the guideline structure in the left hand side window. The data value itself is selected in the top right pane and the complete error message is shown in a dialogue box.
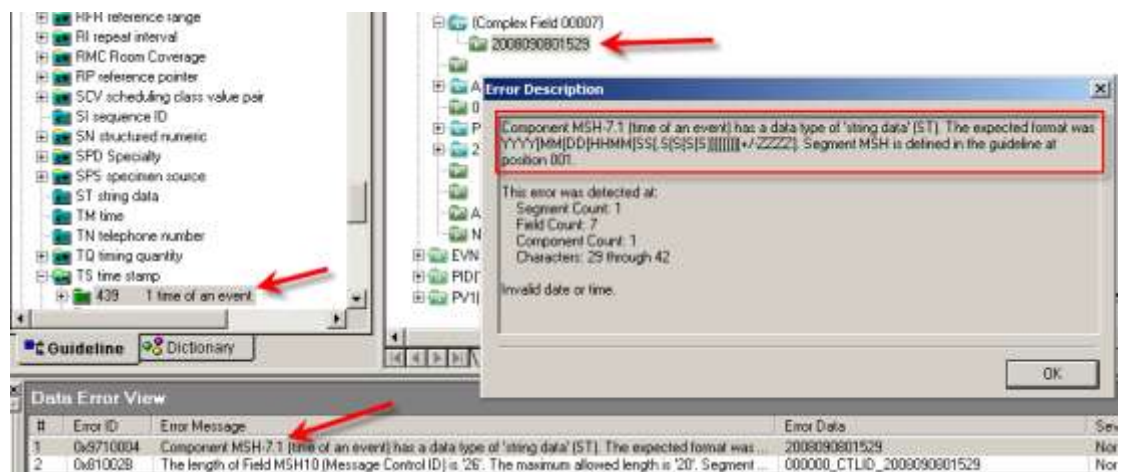
In this case the data value is incorrect – it is 13 characters in length where the specification expects it to be an even number of characters up to 14, followed by a period followed by up to 4 digits followed by a plus sign followed by further 4 digits – a standard timestamp format with optional trailing components. We have 13 characters in the value so the value is not valid.

We can do one of two things:
1. Modify the value by truncating a trailing digit (losing seconds precision) or appending a digit (perhaps add a trailing zero to increase precision of seconds)
2. Modify data type so that any string is accepted as valid

In the first instance we must consider whether the source system, from which this data is assume to have come, can be modified to send valid dates. If it can, then this would be a correct approach. If it cannot then we must modify the guideline to accept dates as given and then deal with the issue in the SOA layer.

Let's assume we cannot modify the sending system and we must modify the data type.

As installed, the B2B Document Builder global property settings prevent modification of base components in guidelines; these are things like data types and lengths. To allow us to make necessary modifications we must change the relevant properties to relax the rules.

Click Tools → Options. From the dropdown next to Common→Guideline View→Usage Restriction→Modification select All. Click OK to dismiss the dialogue box.
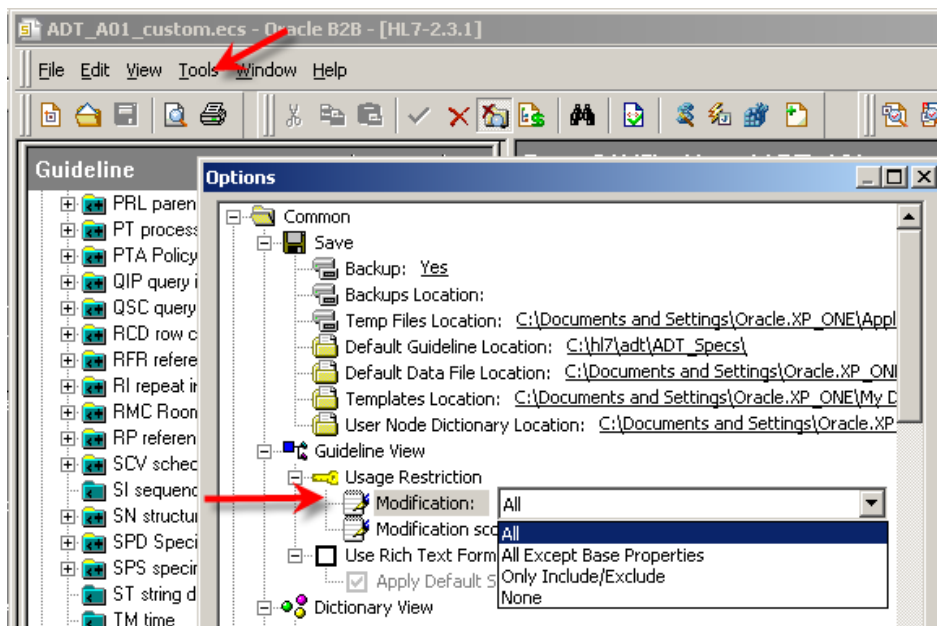
We are ready to modify the guideline structure to better fit our data.

In the Analyzer / Data pane click the parent node of the date/time value in error. Note in the left hand pane that the corresponding data structure node was selected. Click the Properties sub-tab immediately below the Analyzer / Data pane.
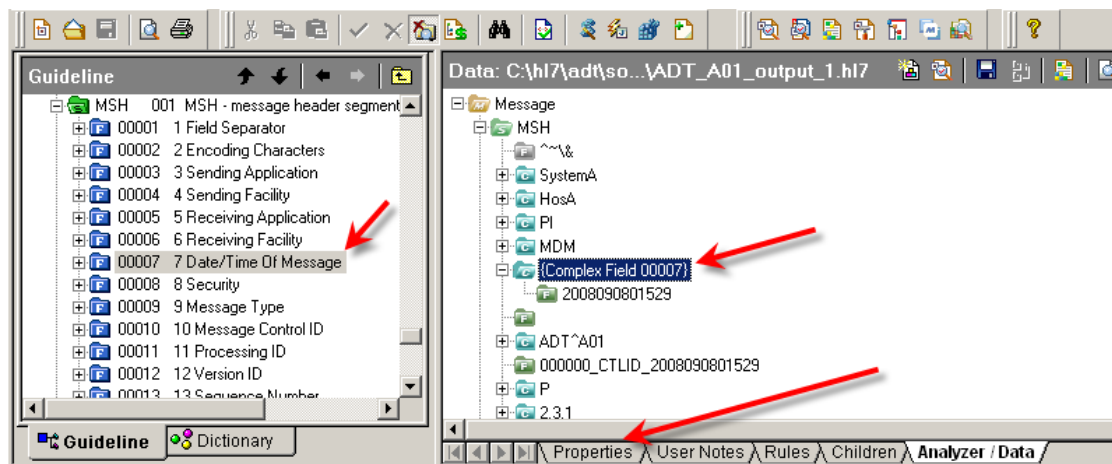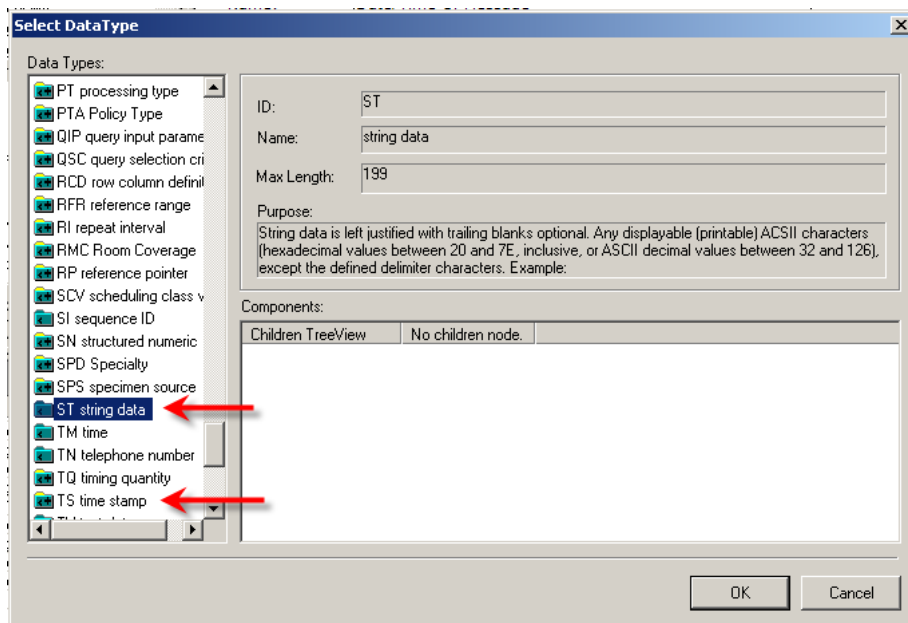

Properties of the node MSH-7 will replace the Analyzer / Data in the right hand pane.

Since we are going to change the data type associated with MSH-7 in this guideline click the ellipsis button to the right of the data type text box.
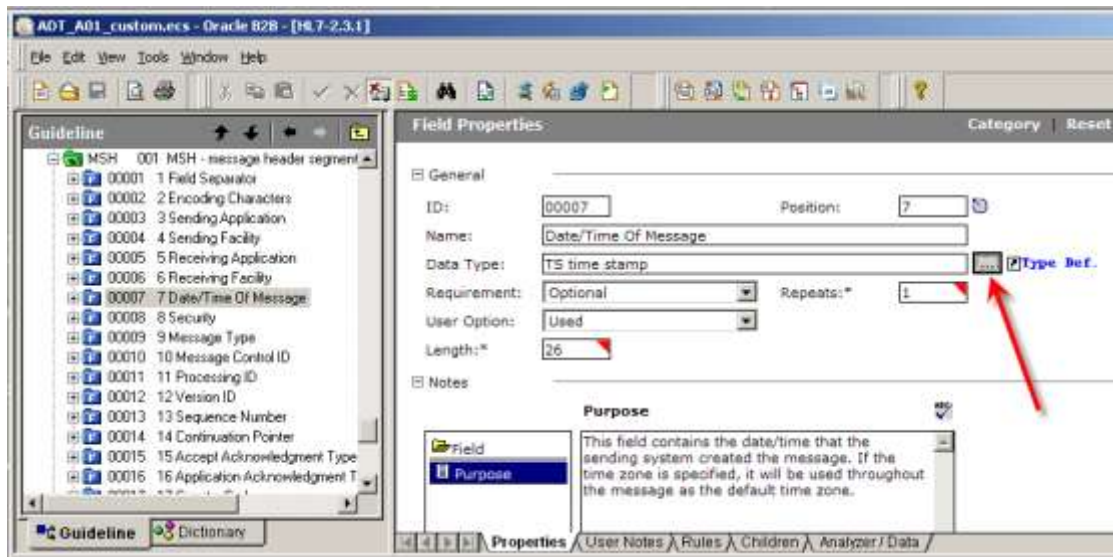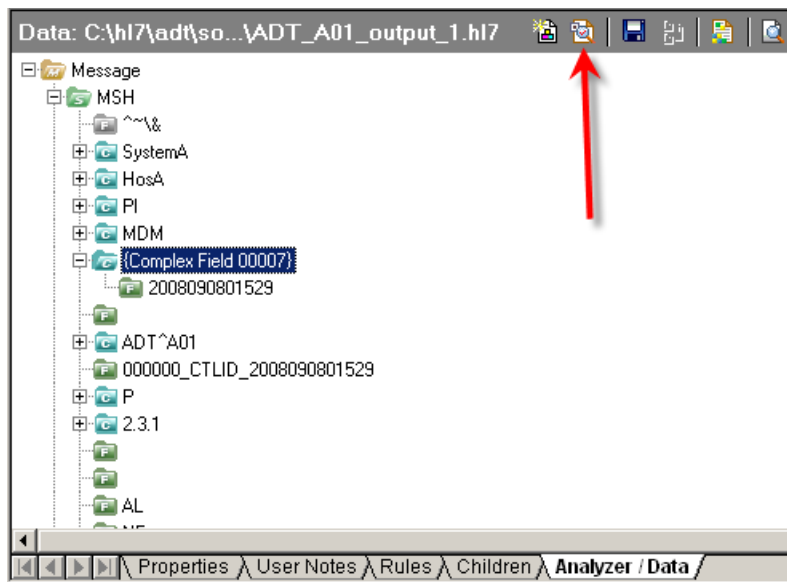

When the dialogue box opens the TS (Time Stamp) data type is selected in the list at the left. Select ST (string) and click OK.

Note the change in the MSH-7 properties. Click the Analyzer / Data sub Tab to switch back to data view. We will analyze data again to see the effects of the change.
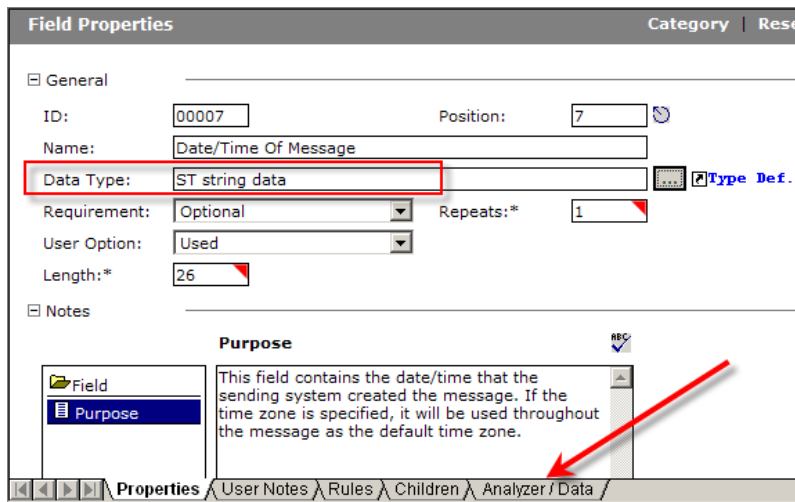
Click the Analyze button.

Click Next, Next, accepting defaults.

The MSH-7 data error is no longer there. The data type change caused the data present in MSH-7 to be considered valid. Let's inspect other errors. We notice two more errors of the same kind as the one we just addressed, error 2 and error 8.

Let's repeat the steps to change data type of EVN-2 and PV1-44 from TS to ST. For each select the parent of the data item in error, click the Properties sub-tab, click the ellipsis to the right of the data type property, select ST (string) data type form the list and click OK.

Repeated analysis shows that the two more errors are resolved.

Let's consider errors 1, 2 and 5. The message says something like:

> The length of Field XXXNN (*description*) is 'NN'. The maximum allowed length is 'NN'.
> Segment XXX is defined in the guideline at position NNN.
>
> This error was detected at:
>   Segment Count: N
>   Field Count: NN
>   Characters: NN through NN
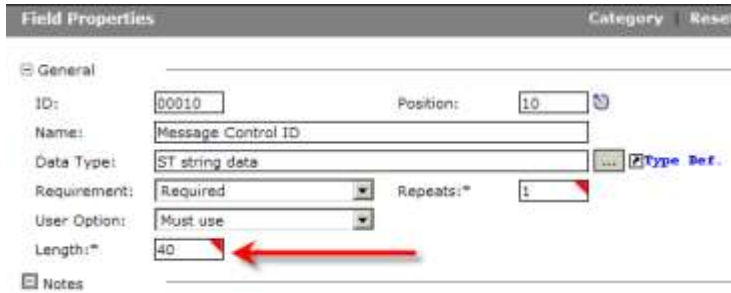>
> The data element is too long.

As before, we have two choices – fix the data of modify the guideline. We already decided that we can't fix the sending system so we can't fix the data. Since these are data lengths we must make judicious guesses on the maximum length of data we are likely to get for the specific fields. In this case we have fields MSH-10 (Max Length 20), PID-3 (Max Length 20) and PV1-19 (Max Length 20). Error messages say that data lengths are 26, 22 and 23 respectively. To sidestep the issue let's make all of them 40 characters in length.

Proceed as follows: Double-click error N message, dismiss the dialogue box, click the Properties sub-tab.

Change max length from 20 to 40. Repeat for the next error.

Save the modified guideline and repeat analysis – click Analyzer / Data, click
Analyze, click Next, click Next.


Three more errors are gone.

The first kind of errors we resolved was data format errors. The second kind of errors
we resolved was data length errors. The errors we are left with are invalid value
errors. Looking closely at the error messages leads us to believe that "I" is not a valid
value for PV1-4, MAIN is not a valid value for PV1-7.14 and VISIT is not a valid
value for PV1-19.5. The guideline and the analyzer together validate data against the
list of values where lists of values are provided in the guideline.

Data Error View

| #	Error ID	Error Message	Error Data
1	0x810024	Field PV14 (Admission Type) does not contain a valid identification code: 'I' is not allowed. Segment PV1 is defined in the guideline ...	I
2	0x810062	Component PV17-14 (identifier type code) does not contain a valid identification code: 'MAIN' is not allowed. Segment PV1 is define...	MAIN
3	0x810062	Component PV119-5 (identifier type code) does not contain a valid identification code: 'VISIT' is not allowed. Segment PV1 is define...	VISIT

Click on the error 1 description, click on Properties sub-tab, expand PV1-4 node in
the guideline tree in the left hand pane and select Admission Type component.
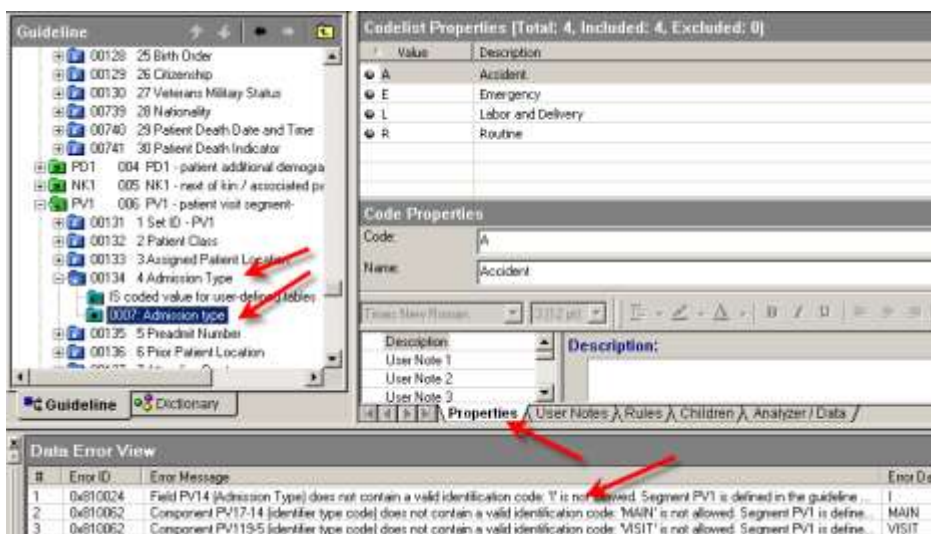
Note the top right pane displaying a code list, with all valid values. Here too we have a choice of fixing the source system or adding values to the list. We will add values to the list. Click the New Code button and add code "I" with name "Inpatient". Click Apply button.
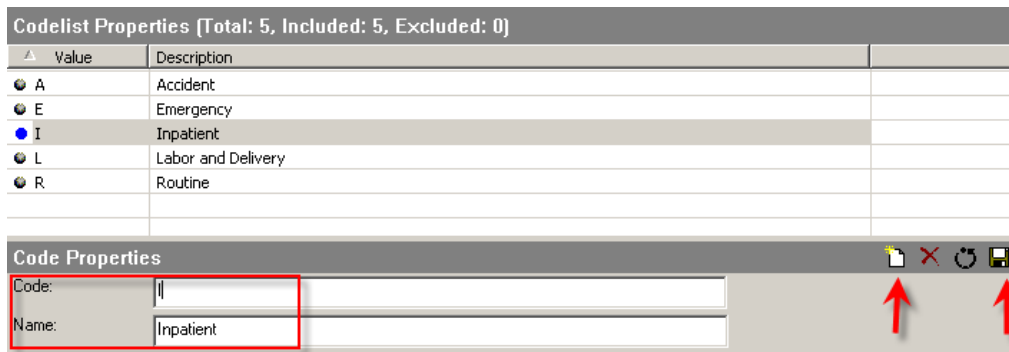
Repeat the process for PV1-7.14 adding MAIN to the code list.
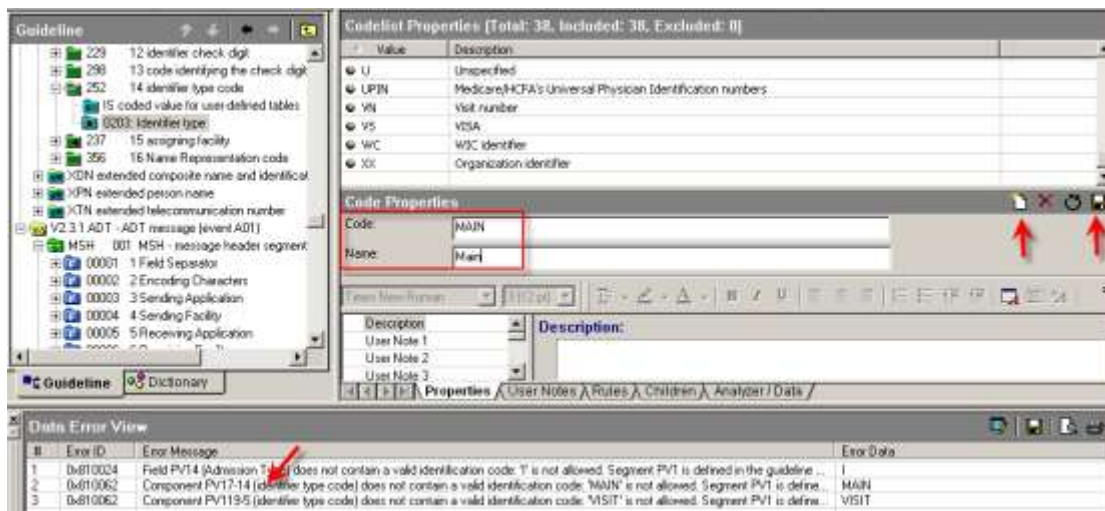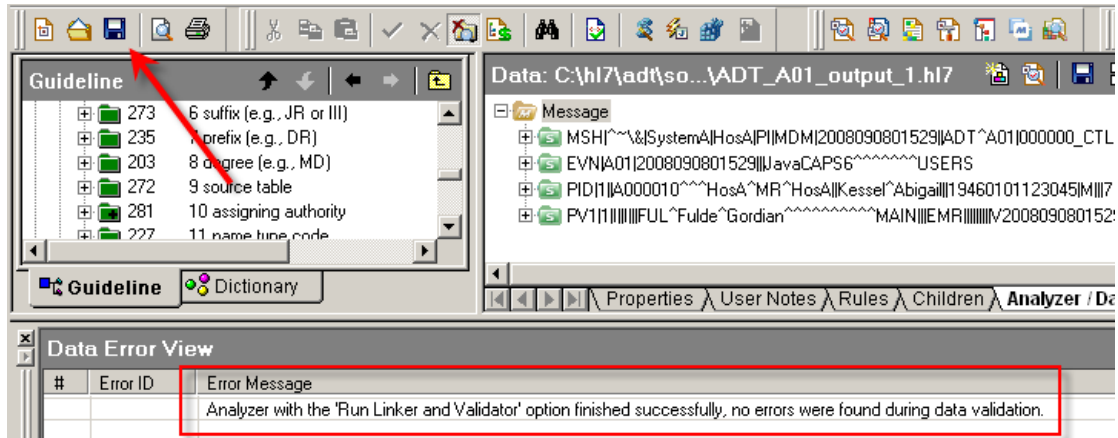

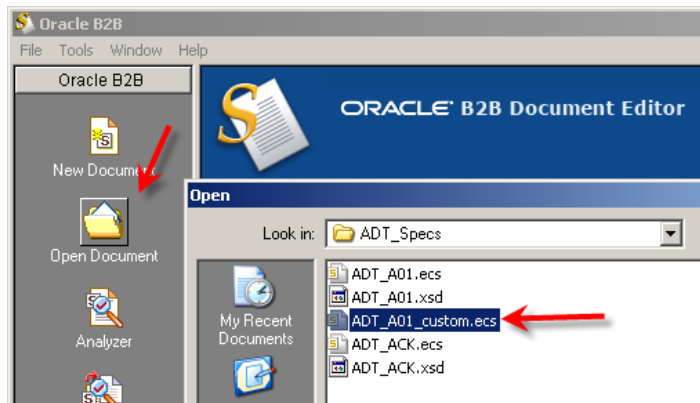Repeat the process for PV1-19.5 adding VISIT to the code list.
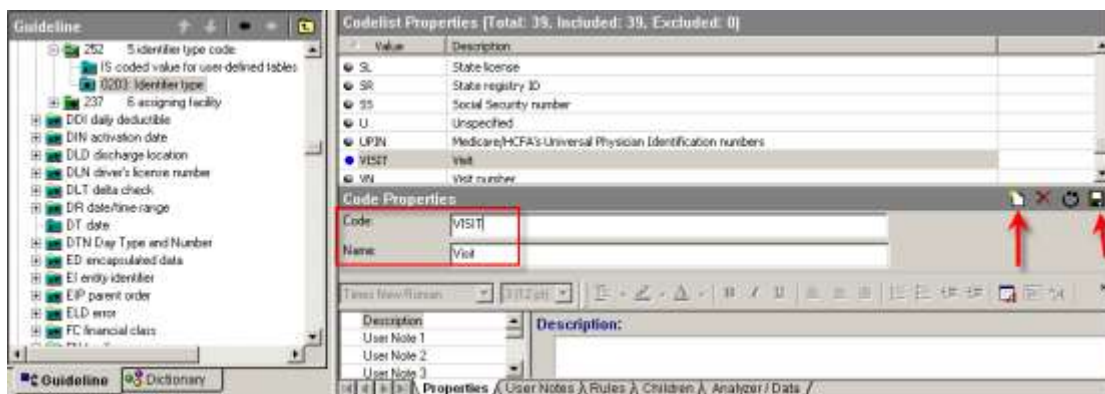
Click Analyzer / Data and repeat analysis.

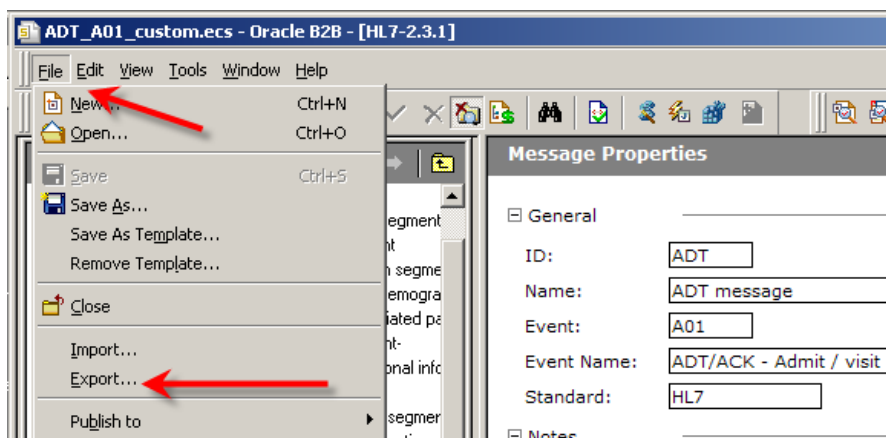Analyzer is happy. Save the modified guideline.
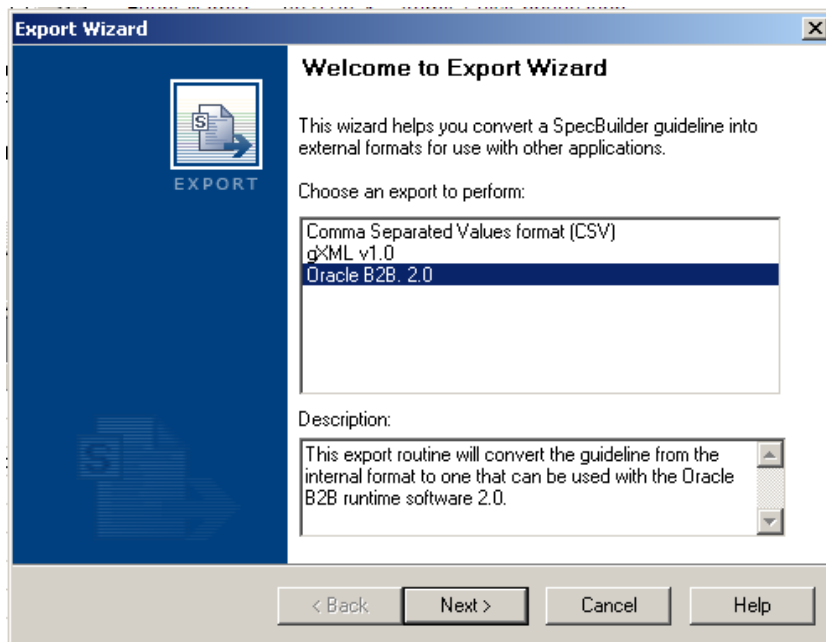
Close the Data Analyzer window.

Back in B2B Document Editor click Open Document and locate the modified ADT_A01_modified.ecs guideline file.
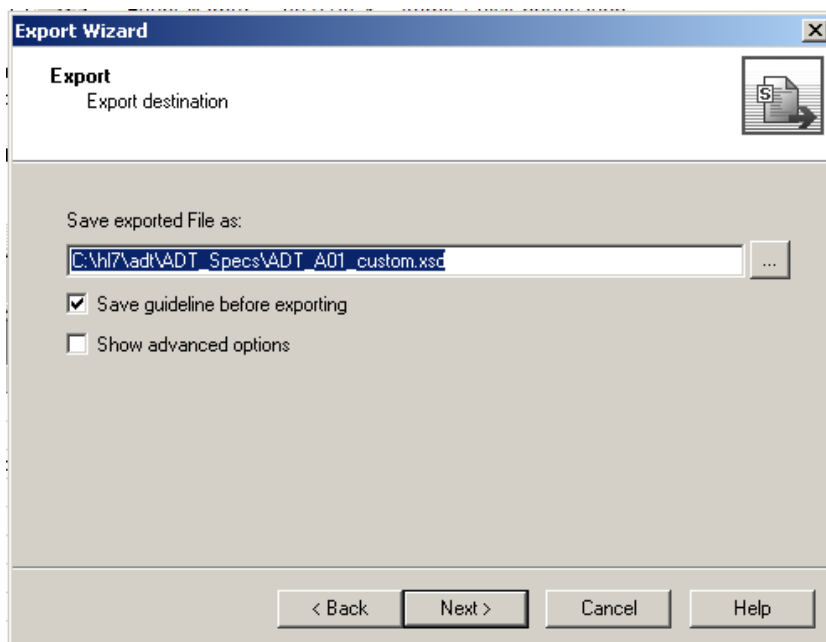

Export the guideline to generate the XML Schema Document that corresponds to the modified guideline.

As before, choose Oracle B2B and click Next.

Specify C:\hl7\adt\ADT_Specs\ADT_A01_custom.xsd as the XML Schema Document's name and location, then click Next and Finish.

Close the B2B Document Editor. We are done with it.

We exported the customized HL7 v2 ADT A01 definition in the XML Schema form which would be used in the SAO Layer.

Modify Document Definition in B2B

Start the B2B Trading Partner Manager console by pointing the web browser at http://localhost:7001/b2b. Log in as weblogic/welcome1.

Click the Administration link, click the Document Tab, expand HL7 node tree all the way to ADT_A01_DocDef. Click ADT_A01_DocDef and update document definition and transaction set files. Click Save to save changes.
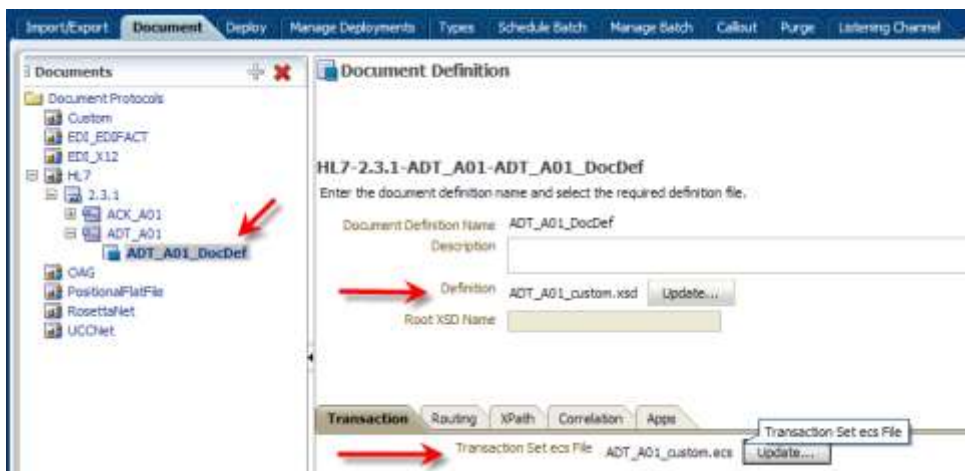
Save and deploy all Trading Partner Agreements which use the modified document definition. If you followed all articles so far these will be HL7ReceiverOuboundTPA_Agr outbound agreement and HL7ReceiverTPA_Agr inbound agreement.
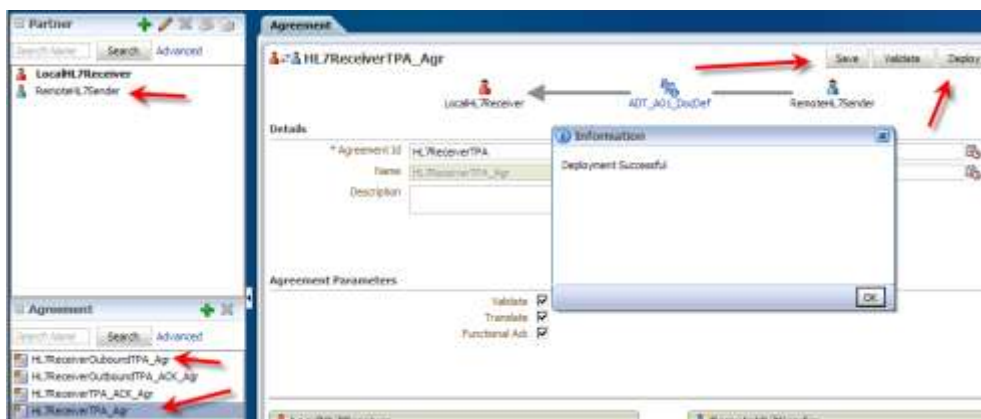
Note, in the screenshot above, that the Validate property on the inbound agreement is checked. The message will be validated against the agreement. If what we did so far works, the message, which in previous articles failed to validate, will now be accepted as valid.

File Writer Solution

We modified the guideline file and the resulting XML Schema Document is different. The differences are in restrictions and value lists, not in structure. We expect that unless the B2B Adapter or the Mediator performs schema validation on incoming data the differences will not be significant and the file writer solution will work without changes.

Exercise HL7 Inbound solution

Start the B2B Trading Partner Manager Web Console, http://localhost:7001/b2b, log in and click the Reports link. Note that there are no messages shown. No messages have been received yet.

Open a command window and run the HL7Browser:

```
C:\jdk1.6.0_20\bin\java.exe -jar C:\tools\Hl7Browser.1.0\HL7Browser.jar
```

When the UI appears click the "Open an HL7 File" button, locate the ADT A01 transaction file, C:\hl7\adt\sources\ ADT_A0x_output1.hl7, and open it.
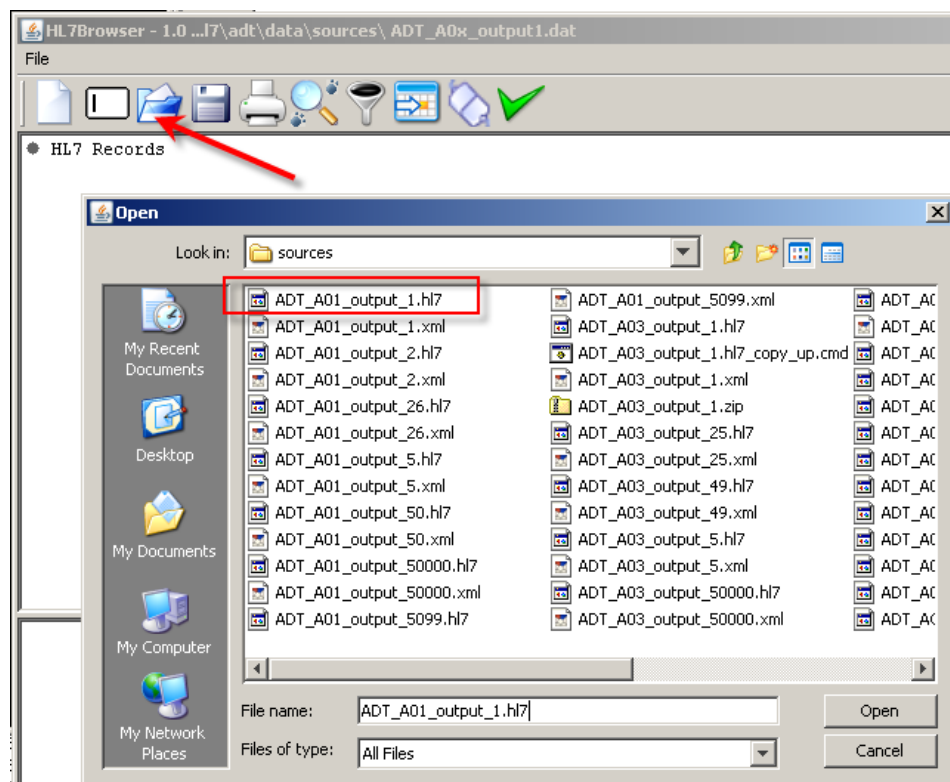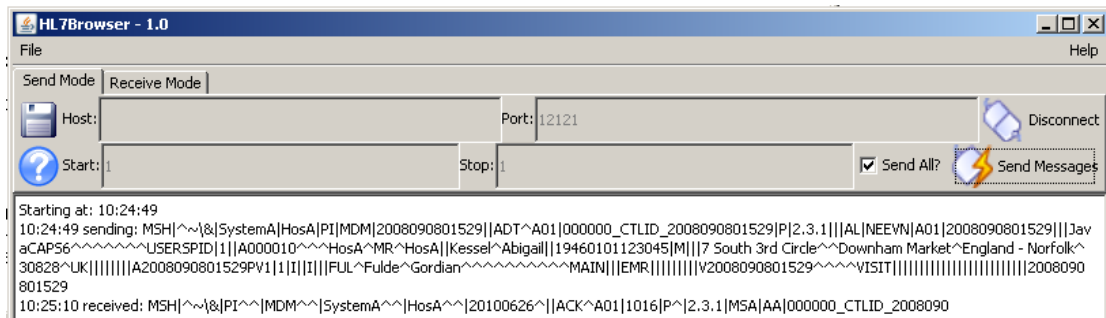

Click "Run the network utility" button, provide localhost as the Host and 12121 as the Port (recall this is the port on which the B2B listener is listening) and click Connect.
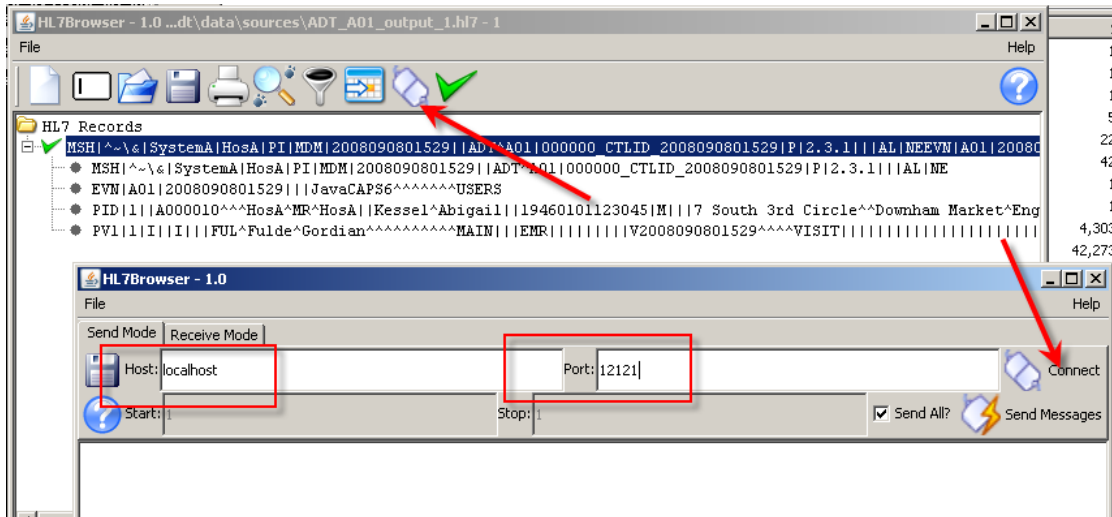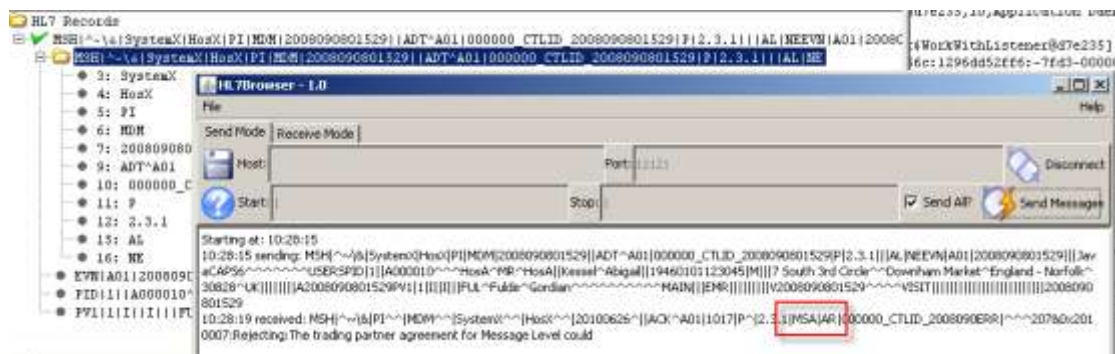
Once connected, click the "Send Messages" button.

Observe message exchange. The message was sent and the ACK received.

The message was valid.

Let's modify the message so that it originates from SystemX HosX, rather than SystemA HosA as expected. Disconnect, re-connect and send the message again.



Message has been rejected – MSA-2 == AR.

Let's restore the message sender to the original SystemA HosA but change the PID segment ID to PIX. Disconnect, connect and Send Message again.
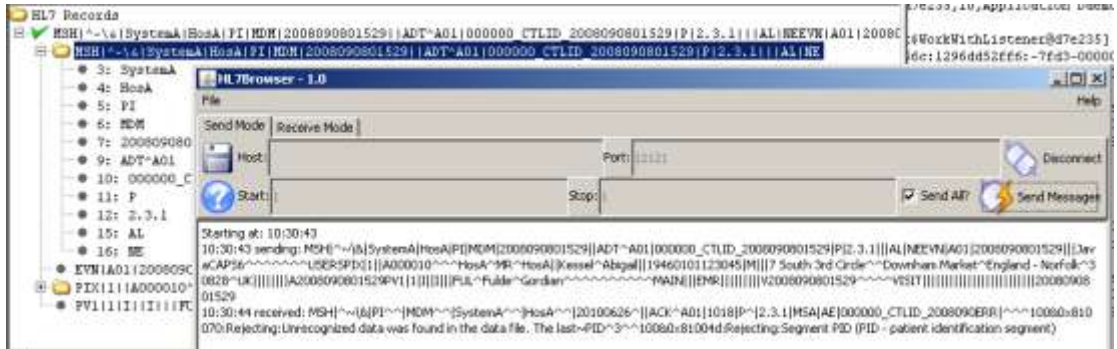
This time we have MSA-2 == AE, Application Error. Message was rejected for structural reasons "`ERR|^^^100&0x810070:Rejecting:Unrecognized data was found in the data file. The last~PID^3^^100&0x81004d:Rejecting:Segment PID (PID - patient identification segment)`".
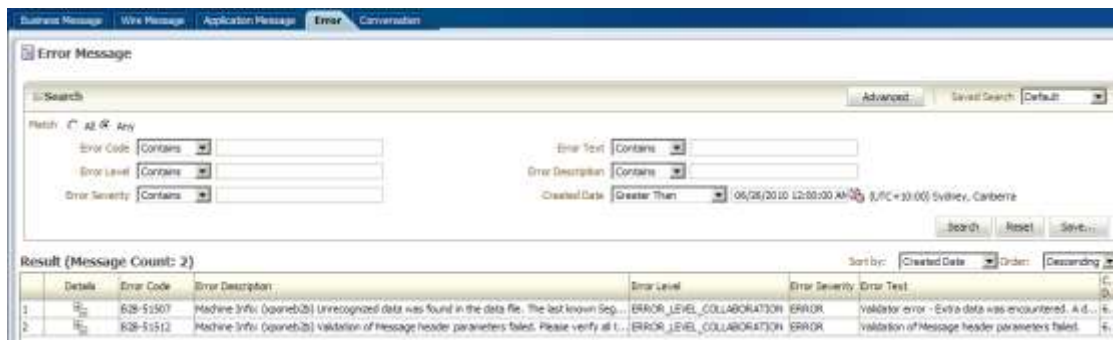
Valid and Erroneous Message Tracking

Start the B2B Trading Partner Manager Web Console, http://localhost:7001:b2b, Click Reports Tab and click Search.
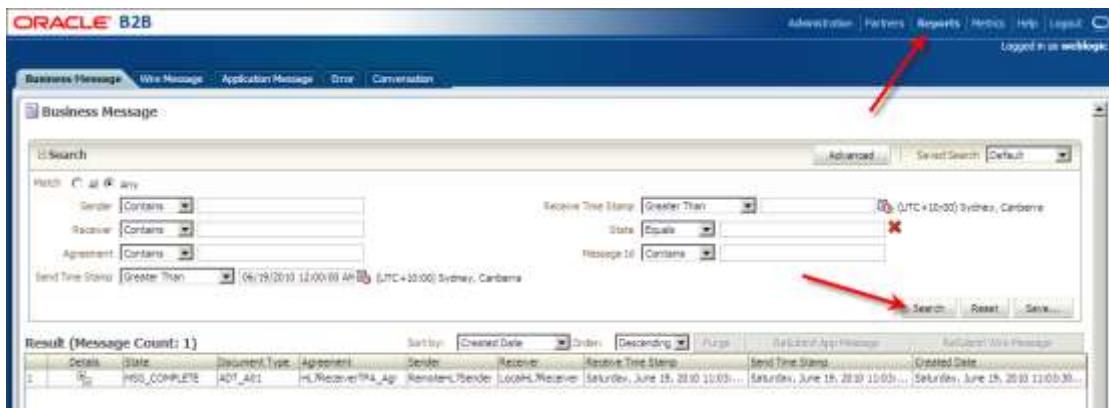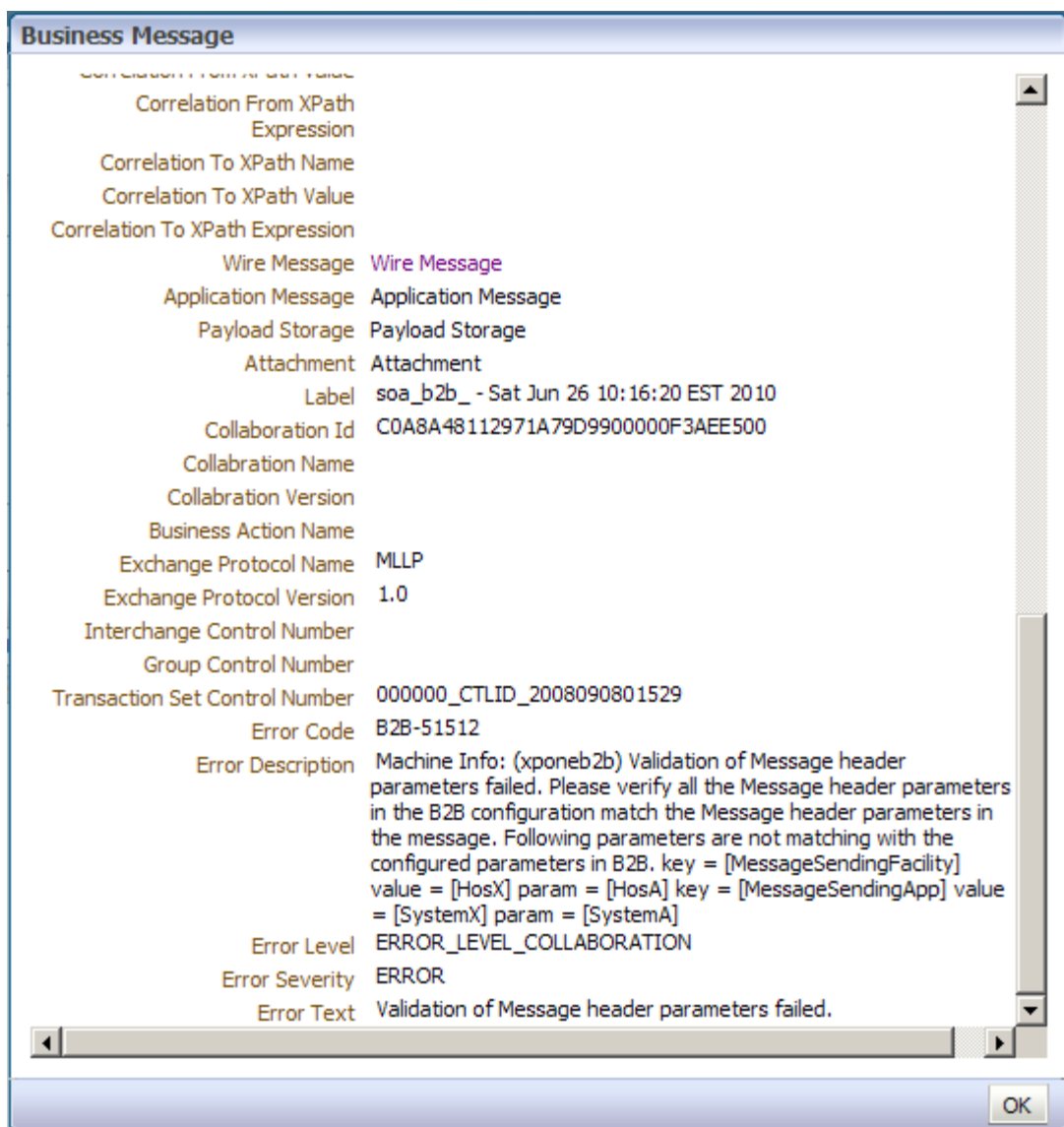

We see 6 Business Messages. One for each message we sent – one valid two invalid, and one each for the acknowledgments to each message.
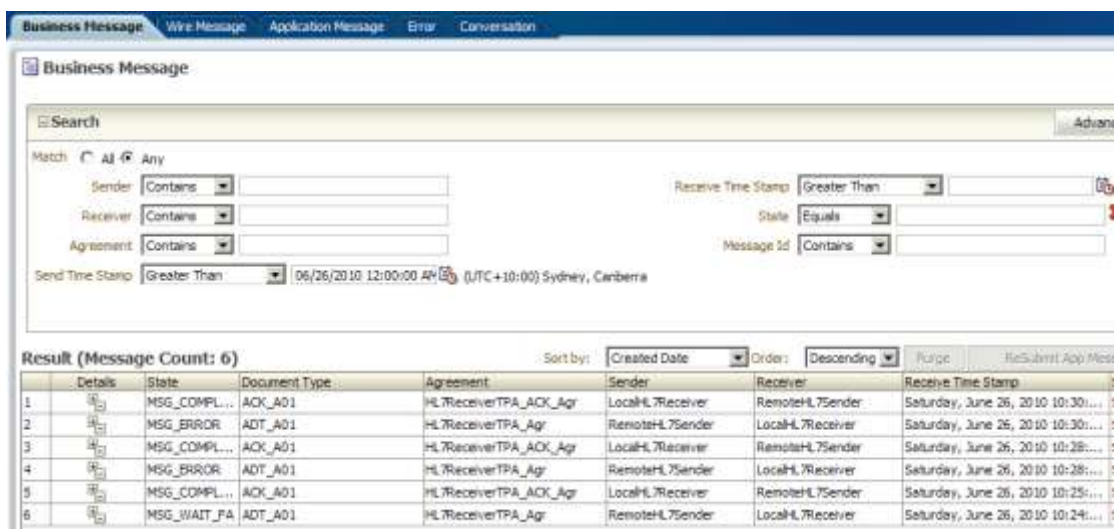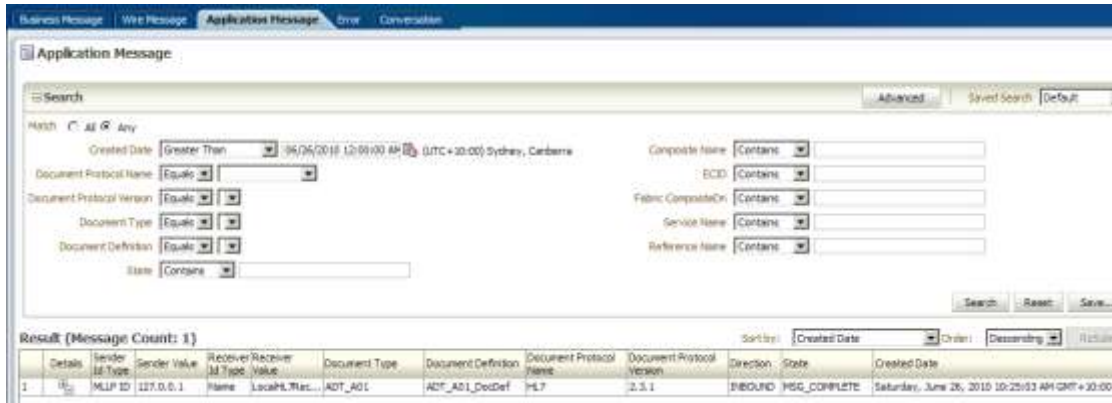
In the Error tab we see two messages with error text shown.
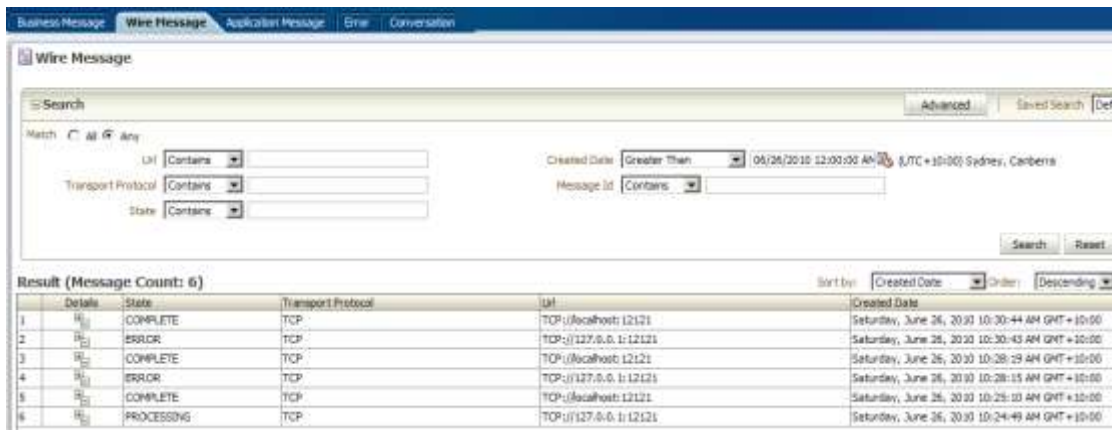

Clicking Details button for one of the messages allows us to drill down into details of the error, then drill further into payload and wire message.

Clicking Application Message Tab shows only valid messages. There is only one in this exercise.

Wire Message Tab shows message State, Protocol and End-point URL at a glance.

Metrics link shows a dashboard with graphs showing message errors, completions and processing times, tabulated view of messaging and tabulated summary of errors. The screen capture does not give justice to the display.
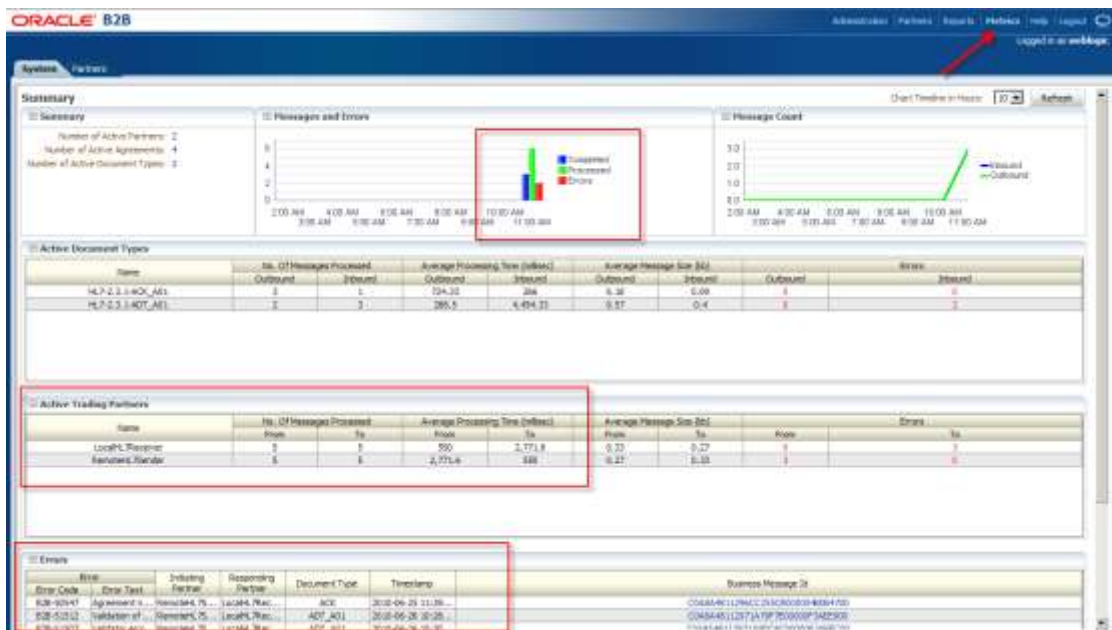
This concludes the exercise.

Summary

Oracle SOA Suite B2B component can be used to provide HL7 v2 messaging support for healthcare environments.

In this article we created a customized HL7 v2 ADT A01 structure which allowed us to successfully validate incoming messages. We modified the document definition and the Partnership Agreements to use the custom structure and validate messages as they came in.

The customization we discussed here only scratches the surface of what is possible with the Oracle B2B Document Editor.

References

[1]   Oracle B2B Site, Available:
        http://www.oracle.com/technology/products/integration/b2b/index.html,
        Accessed: 4 June 2010

[2]   Oracle B2B User's Guide, Available:
        http://download.oracle.com/docs/cd/E15523_01/integration.1111/e10229.pdf,
        Accessed: 19 June 2010